Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 10 (2013) 138 – 146

Procedia Technology

International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013

Chaos based Edge Adaptive Image Steganography

Ratnakirti Roy[a,]*, Anirban Sarkar[a], Suvamoy Changder[a]

[a]*Department of Computer Applications, National Institute of Technology, Durgapur,713209, India*

Abstract

Steganography is the science of hiding data into innocuous objects such that the existence of the hidden data remains imperceptible to an adversary. Steganography in images have varied techniques of implementation developed over time. Protection of the hidden information from an adversary is the most important goal of steganography and hence it is obvious that the security of a steganography system will increase if the payload remains illegible to an attacker even if he holds knowledge about the embedding method. It is also evident that certain areas in an image are more efficient for hiding data than the other parts of the image. These are called Regions of Interest or *ROIs*. Edge areas in an image are one of the ROIs that can be used for steganography. This paper proposes an edge adaptive image steganography mechanism which combines the benefits of matrix encoding and LSBM to embed data and also uses a chaotic mapping scheme to provide enhanced security to the payload. Efforts have been given to ensure that the proposed mechanism conforms to high Imperceptibility and Fidelity which are the essential quality requirements for any image steganography system.

Selection and peer-review under responsibility of the University of Kalyani, Department of Computer Science & Engineering

Keywords: Image Steganography; Object Oriented Steganography; Edge Detection; Chaotic Maps; LSB Matching; Matrix Encoding

1. Introduction

In the recent years, the use of internet services has become more pervasive and affordable than ever before. As a result, today millions of people communicate over the internet and huge volume of data transfer takes place via the

* Corresponding author. Tel.:+91-9434379777.
*E-mail address:*rroy.nitdgp@gmail.com

Selection and peer-review under responsibility of the University of Kalyani, Department of Computer Science & Engineering
doi:10.1016/j.protcy.2013.12.346

plethora of services offered by the web. Despite the role of internet as an excellent worldwide publicized medium for data transmission and sharing, confidentiality of information over the internet demands a lot more. Data over internet may be stolen, intercepted, illegally modified, *anonymized* [1] or even destroyed by an adversary resulting in intellectual property rights infringement, data loss, data leakage and data damage. Hence, it is vital to maintain the privacy and confidentiality of data during its transit through the internet. To preserve the privacy and confidentiality of important data over the internet it must be provided with a metaphorical envelope such that its contents are revealed only to the intended receiver. Data hiding techniques such as *steganography* precisely aims at performing this task.

Steganography is the science of hiding information into innocuous objects in a manner such that the existence of the secret information remains imperceptible to an external adversary. The innocuous object itself is called the *cover* and the hidden information it the *payload.* The choice of cover ranges from simple text files to images, audio and video. Images are widely used as a cover for steganography because of its pervasiveness in daily applications and high redundancy in representation. Image steganography techniques are classified into *spatial* and *transform domains*. Spatial domain methods apply direct pixel manipulation for information hiding. They are also characterized by *simplicity*, *shortened implementation time*, *reduced hardware requirement* and *overall low time complexity* [2]. All image steganography techniques, irrespective of their domain of implementation must focus on three vital points-*where to hide the information in the image, how safe is the embedding* and *how secure is the payload in case of exposure to an adversary*. Many algorithms for image steganography exist and each of them addresses these differently.

Embedding secret data into specific *regions of interest (ROI)* within an image is a relatively new approach to image steganography. *ROI*s may be any object in the image that produces *least amount of distortion* when embedded with data. One such *ROI* is the edge regions in an image. Edge regions are suitable for data hiding because the human visual system is less sensitive to distortions in edge regions and it also provides *randomized pixel positions*. Embedding into randomized pixel positions scatters the payload throughout the cover and reduces the probability of detection by steganalyzers [3]. *ROI* based steganography can be considered to be a suitable alternative for development of better steganography algorithms [4]. On the other hand, there are varied approaches to ensure the security of the payload. Common approaches include pre-encrypting the payload or distorting it in a *pseudo random manner* using various mathematical transforms and mappings.

This paper aims to present an ROI based spatial domain steganography mechanism that embeds secret data into *edge regions* of a cover image. In order to enhance the security of the payload it is subjected to a pre-embedding distortion using a *chaotic mapping technique* which ensures that the actual secret message does not get revealed even on exposure to an adversary having knowledge about the embedding mechanism.

2. Related Research

Edge region based embedding falls under the category of *Object Based Steganography* and some interesting research work has been carried out in developing edge based image steganography techniques. At the same time, methods for utilizing the *chaotic maps* for image encryption have also been proposed. Most of the edge based schemes emphasize on the *Pixel Value Differencing* (*PVD)* [5] to distinguish between edge and smooth pixels. On the other hand, *chaotic map* based schemes have utilized a variety of chaotic maps like *Henon's Map, Logistic Map* and so on. Few of the important research works relevant to the current context are presented next. In [6] a PVD based steganography technique is proposed. The proposed method is an improvement on the Least Significant Bit Matching Revisited (LSBMR) technique. The classification of the pixels determines the amount of data that can be hidden within that pixel. The smooth region pixels hold lesser amount of data while the edge pixels hold more embedded data. Similarly, an edge based embedding scheme proposed in [7] uses a *Laplacian detector* to find edge pixels and embeds into the sharper edges using the LSB Replacement (LSBR). The smooth region pixels are however left undisturbed. A LSB steganography method using PVD providing a larger embedding capacity and imperceptible stego images is proposed in [8]. The authors claim that the proposed method is superior to Wu et al.'s PVD based scheme [5].

R.L Tataru et al., [9] have proposed a spatial domain chaotic map based steganography scheme which uses PVD for pixel pair differentiation and a chaotic mapping scheme to choose the two pixel group for embedding and

extraction. In [10], an image scrambling technique using chaotic cat mapping has been proposed. The image is initially distorted using the cat chaotic mapping and then XOR operation is performed between certain pixel value of the digital image and a chaotic value. For restoration of the image, an inverse permutation is performed. A symmetric cipher proposed in [11] uses the 2D Standard map and 1D logistic map to encrypt colour images. The scheme employs two kinds of diffusion processes which are completed by mixing the properties of horizontally and vertically adjacent pixels using a Logistic map, respectively. Similarly, a watermarking system proposed in [12] uses *Lorenz Map* and Arnold Cat Map to encrypt the payload and in order to spread the watermark signal in all the regions of host image chaotically. Apart from these, there are other image ciphering techniques which use chaotic maps like the Heron's map [13] for scrambling images for secure transmission through a network.

3. Proposed Mechanism

The proposed mechanism works on the spatial domain and uses *Canny's Edge Detection* [14] for locating the edge pixels in the cover image and hide data into the selected pixels. Canny's method has high immunity to noise and can detect true weak edges [15]. It has been considered to be an optimized and standard method for detecting edges in images as compared to other techniques of edge detection [16, 17]. Edge adaptive embedding provides necessary *pixel randomization* as edge regions are scattered more or less throughout the cover image.

In an effort to increase the security of the payload even in the case of exposure to an adversary, Cat Mapping [18] is applied to distort the payload initially. An interesting feature of Cat mapping is that if it is applied to an image, after a certain number of iterations the mapping returns the original image. The intermediate stages are however distorted and bear negligible resemblance to the original image. It is an *area preserving* map and the basic operation of the map is that the image is sheared one unit up, then one unit to the right, and all that lies outside that unit square is shifted back by the unit until it's within the square. This can be mathematically expressed as

$$\Gamma\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \begin{bmatrix}1 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} mod\ 1 \tag{1}$$

where $\Gamma\left(\begin{bmatrix}x\\y\end{bmatrix}\right)$ gives the Cat Map transform over the original pixels x and y. Generally for an NXN image,

$$\Gamma\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \begin{bmatrix}1 & p_1\\ p_2 & p_1p_2+1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} mod\ N \tag{2}$$

where $p_1, p_2 \in \mathbb{Z}$ and $p_1, p_2 \geq 1$.

Thus, it is evident that (1) is a special case of (2) when $p_1=p_2=N=1$. The values of p_1, p_2 may be altered to increase the number of iterations required to bring back the original image. The proposed mechanism utilizes Cat mapping in two steps. In the *first step* (performed before embedding), the payload is iterated k *times* ($k<n$, where k^{th} iteration gives the most distorted transformation of the payload and n is the number of iterations required to construct the original image). The *second step* is performed during extraction where the retrieved distorted payload is restored to retrieve the original message.

The actual embedding is performed using a combination of *matrix encoding* [19] and *LSB Matching (LSBM)* [20]. Matrix encoding ensures that the secret message is embedded into the cover image with minimum number of pixel changes and LSB matching alleviates the *Pair of Value* (PoV) effect of the LSB Replacement (LSBR) method. The combined scheme works as follows.

Let a_1, a_2 and a_3 be three modifiable bit positions belonging to pixel components *Red (R), Green (G), Blue (B)* respectively and let x_1 and x_2 be two message bits which needs to be embedded making at most one change in the target bits. Now, consider the following:

$$x_1 = a_1 \oplus a_3 \tag{3}$$
$$x_2 = a_2 \oplus a_3 \tag{4}$$

All possible conditions (in pairs) that may arise for each of the modifiable bits along with the necessary action (*without using bit-flipping*) to be taken during embedding are listed below.

Table 1. Embedding Condition-Action List

Condition	Action to be taken
$x_1 = a_1 \oplus a_3$ $x_2 = a_2 \oplus a_3$	No change required
$x_1 = a_1 \oplus a_3$ $x_2 \neq a_2 \oplus a_3$	Change component G to match conditions (3) & (4)
$x_1 \neq a_1 \oplus a_3$ $x_2 = a_2 \oplus a_3$	Change component R to match conditions (3) & (4)
$x_1 \neq a_1 \oplus a_3$ $x_2 \neq a_2 \oplus a_3$	Change component B to match conditions (3) & (4)

The proposed steganography mechanism consists of *two* phases, namely, *Phase I* and *Phase II*. The first phase performs the pre-embedding payload scrambling and embeds the distorted payload in the edge regions of the cover while the latter phase extracts the hidden information and descrambles the payload to reveal the original message.

3.1. Phase I

3.1.1. Payload Scrambling Algorithm

The payload scrambling algorithm works by finding the k^{th} iteration of Cat mapping on the input image which has the lowest similarity with the original payload. It then applies k successive cat map based transforms to generate the distorted payload. The detailed algorithm is as under:

Input: Original payload *S*
Output: Scrambled payload *T*
Algorithm:
Step 1: Find the number of Cat map transforms required to distort the payload and regenerate the original again. Let it be denoted by *p*. At each iteration *i (i<p),* find the two dimensional correlation coefficient between *S* and the output of the corresponding iteration. Store correlation coefficients in an array *A*.
Step 2: Find index of *minimum(A).* Let it be denoted by k_{max}.
Step 3: Set *rem:=p-*k_{max}. Perform cat map transform k_{max} *times* on *S* to generate the distorted payload *T*.

3.1.2. Embedding Algorithm

The embedding algorithm utilizes the combines matrix encoding and LSB matching to embed the distorted payload into the cover image *I*. Once the payload has been embedded, the generated stego image can be sent to the receiver for extraction.

Input: Cover Image *I,* Scrambled payload *T*
Output: Stego Image
Algorithm:
Step 1: Find the edge pixels in the cover image *I* using an edge detection algorithm.

Step 2: Convert the payload to its binary equivalent. Let this be denoted by *B*.

Step 3: Calculate the length of the payload. Let it be denoted by *L*.

Step 4: Calculate the number of pixels needed for embedding. Let this be denoted by *pixnum.* Set *pixnum:=L/2*.

Step 5: Set a counter *i*. Set *K=1*.

For *i:=1 to pixnum*

1. Set *pix*:=i^{th} pixel values from the set of pixels selected in step 1.
2. Let *a1*=Red Plane LSB of *pix*, *a2*=Green Plane LSB of *pix*, *a3*= Blue Plane LSB of *pix*.
3. Let *x1=B(K)*, *x2=B(K+1)*.
4. Perform Embedding using checks for conditions mentioned in Table 1. Replace the original pixel components with the changed component values wherever necessary.
5. Set *K=K+2*.

End For

Step 6: Write the modified image matrix to a file.

3.2. Phase II: Extraction

The extraction algorithm is the reverse of the embedding process. The extraction algorithm searches the edge regions of the stego image and extracts the hidden data using the decoding information sent by the sender. The cat map iteration information *rem* generated in *Phase I* acts as the descrambling key. The extraction algorithm is as follows:

Input: Stego Image, Message Length *L,* Edge pixel information, descrambling key *rem*
Output: Hidden Message
Algorithm:

Step 1: Calculate number of pixels to search for hidden data as *pixnum:=L/2*.

Step 2: Set a counter *i*.

For *i:=1 to pixnum*

1. Set *pix*:=ith pixel values from the set of pixels supplied as decoding information.
2. Let *a1*=Red Plane LSB of *pix*, *a2*=Green Plane LSB of *pix*, *a3*= Blue Plane LSB of *pix*.
3. Perform XOR operation on *a1* and *a3* to get the first message bit *x1*. Perform XOR operation on *a2* and *a3* to get the second message bit *x2* hidden in the pixel.
4. Store the message bits as a binary sequence.

End For

Step 3: Modify the sequence containing the binary message bits according to the appropriate type of the intended data. Let it be *D*.

Step 4: Perform descrambling of *D* using *rem* generated in *Phase I* as the key and Cat Mapping to get the original message.

3.3. Complexity Analysis

3.3.1. Time Complexity and Space Complexity

Let the number of pixels in the cover image be *n*. Step-1 of the *embedding* algorithm in *Phase I* is the edge region detection mechanism which searches for edge pixels in the cover image. The time required by the mechanism to detect edges increases with the increase in the number of pixels. Thus it can be said to have a time complexity of *O(n)*. Similarly, the embedding loop iterates *pixnum* times where *pixnum = L/2* and *L* is the secret message length.

Since, $L<<n$ so the time complexity of the embedding algorithm is $O(n)$. In the same manner the time complexity of the *extraction* algorithm in *Phase II* can be determined to be $O(n)$.

In order to determine the space complexity of the proposed steganography technique the data structures whose size varies with the change in the input are taken to consideration. Matrices are used to store the cover image, the stego image and the secret message. If n is the number of pixels in the cover image, then the memory space requirement increases as n increases. Thus, the space complexity of the proposed algorithm is $O(n)$.

4. Experimental Results and Analysis

The algorithms proposed in *Phase I* and *II* of the preceding section have been implemented in MATLAB and tested on a 32-bit, 2.4 GHz single core processor computer. The cover images are standard images of *Tiffany, Baboon, F16* and *Peppers*. The embedding algorithm has been tested for the level of visual distortion produced (*Fidelity*) and *Imperceptibility to statistical steganalysis* and *security of the payload*. The runtime behaviour of the algorithm is measured in terms of the time required to embed the payload of varying sizes. The output of the edge detector depends on the *threshold* value supplied to it. For test purposes the edge detector was subjected to run with thresholds $t_{low}= 0.004$ and $t_{high}=0.01$ so that more edge pixels are revealed.

For the cat mapping scheme, the values p_1, p_2 (Eqn. (2)) are taken to be equal to *1*. The most distortive iteration k_{max} for *Lena* image of size 100X100 was $k_{max}=74$ at which the correlation (r) between the original payload and the distorted payload is minimum ($r=-0.0397$).

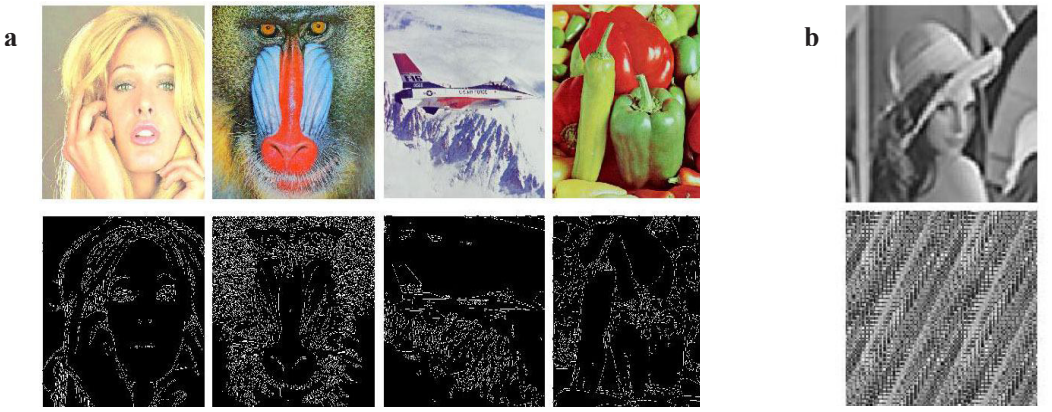

Fig 1. (a) Image showing the cover files and their corresponding edge regions (b) Original payload and distorted payload at iteration k_{max}

4.1. Measuring the Embedding Distortion

The distortion produced in the cover image due to the embedding is measured in terms of *Peak Signal to Noise Ratio (PSNR)* which is calculated using the *Mean Square Error (MSE)*. The quantifiers are defined as follows:

$$MSE = \frac{1}{MXN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij} - y_{ij}\right)^2 \quad (5)$$

$$PSNR = 10\,log_{10}\left(\frac{255^2}{MSE}\right)dB \quad (6)$$

where M, N are the horizontal and vertical pixel dimensions of the cover image, x_{ij} and y_{ij} are the pixel values in the cover and the stego image respectively. In (6), constant value *255* signifies the maximum value that a colour may hold in a pixel having a colour depth of 8 bits. For 24-bit RGB images, each colour component has a colour depth of 8 bits. Higher *PSNR* value indicates better fidelity of the stego image which in turn signifies lower distortion. The *MSE* for RGB images is calculated per colour plane of the image and then the average of the *MSE* of the individual plane gives the Mean Square Error between the cover and the stego image. *PSNR* value is expected to be greater

than *40dB* [2] for high fidelity stego image. The minimum embedding was *8192 bits* for a payload image of size 32 X 32 and the maximum embedding was *80,000 bits* for a payload image of size 100 X 100.

Table 2. Distortion Measure and Embedding Time for various levels of Embedding

Payload Size	PSNR (dB)				Time to Embed (seconds)			
	Tiffany	Baboon	F16	Peppers	Tiffany	Baboon	F16	Peppers
32X32	77.03	76.96	77.05	77.07	0.68	0.69	0.67	0.72
60X60	73.1	73.13	73.13	73.12	0.96	1.09	1.11	1.1
64X64	71.55	71.53	71.5	71.52	1.31	1.25	1.33	1.23
80X80	70.95	70.98	70.98	70.98	1.3	1.38	1.49	1.41
100X100	67.01	67.06	67.1	67.1	2.5	2.87	2.85	2.87
Average	71.94				1.44			

The results in Table 2 indicate that the proposed technique produces high *Peak Signal to Noise Ratio (PSNR)* for all the tested cover images and payload sizes. It embeds at an average rate of *2bits per pixel*. The average *PSNR* for the method is *71.94* which is higher than the minimum threshold for human visual system (40dB). Thus, the proposed mechanism produces *high fidelity* stego image with negligible visible distortion.

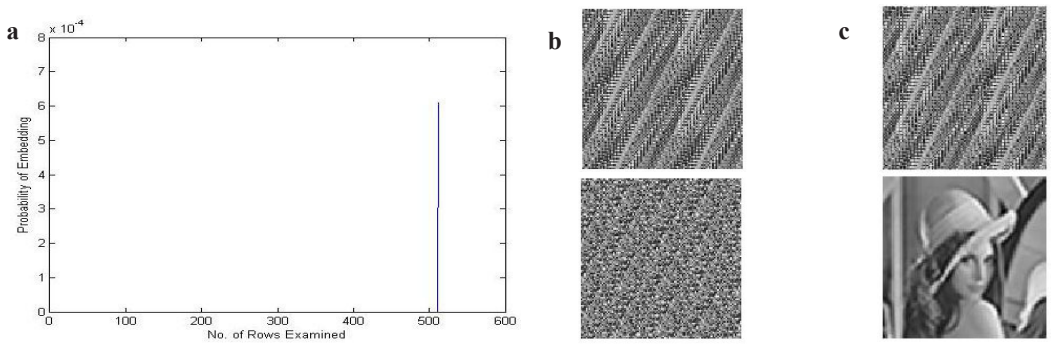

Fig 2. (a) Performance of χ^2-test against the proposed method; (b) Descrambling at $rem \neq p\text{-}k_{max}$; (c) Descrambling at $rem = p\text{-}k_{max}$

4.2. Imperceptibility to Statistical Steganalysis and Security of the Payload

The stego image generated after *Phase I* was tested against χ^2-steganalysis to check for the degree of *imperceptibility* of the proposed method. The highest probability of existence of hidden data in the tested stego image is found to be 6.0976×10^{-4} (Fig. 2(a)). Thus it is evident that the proposed steganography mechanism does not alter the apparent random distribution of the pixels of the cover image even after embedding. It also signifies that the *Pair of Values (POV)* effect has been alleviated and hence the method exhibits *high imperceptibility* to steganalysis techniques which check for the degree of randomness in images to detect existence of hidden data.

The distorted payload (Fig. 1(b)) after extraction has been subjected to the descrambling mechanism mentioned in 3.2. The results (Fig. 2(b)-(c)) indicate that the distorted payload regenerates the original payload *only* when the descrambling key *rem* is same as that was generated during the initial phase of embedding. For any other value of *rem* the payload remains distorted.

4.3. Runtime Behaviour

The proposed mechanism was tested with standard images as covers and a payload of varying sizes (Table 2). It shows a decent performance in terms of the embedding time. The minimum time required for embedding was *0.67* seconds (F16 image) for a payload of size 32 X 32 and the highest time requirement was 2.87 seconds (Baboon and

Peppers images) for a payload of size 100 X 100. However, the average time for embedding clocked to *1.44* seconds.

5. Comparison with other methods

In order to evaluate the performance of the proposed algorithm with respect to the other methods, it is compared with a few of the popular algorithms already available. The comparison is done on the basis of *Fidelity*, *Embedding Time*, *Imperceptibility to Statistical Steganalysis* and *Security of Payload*. The comparative test result (*PSNR* and *Embedding Time*) is listed in Table 3.

Table 3. Comparative data for different algorithms

Cover	Algorithm	Proposed Method		PVD[5]		LSBR	
	Payload Size	PSNR (dB)	Time (seconds)	PSNR (dB)	Time (seconds)	PSNR (dB)	Time (seconds)
Tiffany	32X32	77.03	0.68	60.89	2.43	75.74	0.3
	60X60	73.1	0.96	61.73	2.46	71.87	1.34
	64X64	71.55	1.31	61.13	2.74	70.26	1.55
	80X80	70.95	1.3	60.88	2.74	69.69	1.7
	100X100	67.01	2.5	58.62	3.61	65.84	2.15
Baboon	32X32	76.96	0.69	60.78	2.83	75.75	0.29
	60X60	73.13	1.09	60.81	2.55	71.78	1.37
	64X64	71.53	1.25	60.86	2.61	70.28	1.55
	80X80	70.98	1.38	60.75	2.56	69.68	1.7
	100X100	67.06	2.87	58.43	3.59	65.83	2.2
F16	32X32	77.05	0.67	60.92	2.67	75.74	0.31
	60X60	73.13	1.11	60.93	2.47	71.86	1.29
	64X64	71.5	1.33	60.91	2.57	70.27	1.57
	80X80	70.98	1.49	60.95	2.5	69.69	1.69
	100X100	67.1	2.85	58.67	3.69	65.83	2.2
Peppers	32X32	77.07	0.72	60.94	2.53	75.76	0.29
	60X60	73.12	1.1	60.94	2.53	71.87	1.2
	64X64	71.52	1.23	60.89	2.48	70.29	1.33
	80X80	70.98	1.41	60.92	2.58	69.73	1.7
	100X100	67.1	2.87	58.66	3.57	65.84	2.2
Average		71.94	1.44	60.48	2.79	70.68	1.40

PSNR: Peak Signal to Noise Ratio, PVD: Pixel Value Differencing, LSBR: Least Significant Bit Replacement

5.1. Comparison based on Fidelity and Embedding Time

The degree of *fidelity* is compared on the basis of *PSNR* values for each of the methods taken into consideration. The proposed method shows a *higher* average PSNR of *71.94* as compared to *60.48* and *70.68* of PVD (*2 bpp* embedding rate) and LSB Replacement (LSBR) respectively. It is evident from Table 3 that the proposed technique shows a consistent high *PSNR* as compared with other algorithms taken. Hence it ensures *high fidelity* of the stego-image which is a vital requirement for any image steganography technique. It is also evident that the proposed technique shorter embedding time duration than the PVD. The time required to embed data, averaged over different cover images are *1.44* and *2.79* seconds for the proposed method and PVD respectively. This is because the proposed technique does not use direct pixel comparison based method for differentiating between edge and smooth region pixels However, LSB Replacement (LSBR) has a slightly lower embedding time than the proposed method.

5.2. Comparison on the basis of Imperceptibility to Statistical Steganalysis and Payload Security

The proposed technique exhibits extremely low probability of detection when subjected to χ^2-steganalysis ($P=6.0976 \times 10^{-4}$) which indicates that the apparent random distribution of the image pixels is not disturbed in a large scale due to embedding method adopted. At the same time it also signifies that the *POV Effect* shown by the LSB Replacement (LSBR) method is no longer present. The *POV Effect* has been alleviated in the current method by application of LSB Matching (LSBM). The embedding efficiency of the technique is also improved using matrix encoding. Moreover, the method does not alter the smooth region pixels so there is no *Step Effect* [21] as in PVD.

Neither PVD [5] nor the LSBR adopt any measure to ensure the security of the payload in the event of its exposure to an adversary whereas the proposed steganography technique provides a chaos based payload scrambling-descrambling system to ensure that the actual hidden message is not revealed to an adversary even if he holds knowledge of the embedding method. Experimental results show that the distorted payload cannot be descrambled to produce the original unless the correct descrambling key is supplied.

6. Conclusion

In this paper, an edge adaptive image steganography technique is proposed which exhibits *high fidelity* and *good imperceptibility* to steganalysis attacks. It combines matrix encoding and LSBM for embedding in the *edge regions* of a cover image. It also uses *cat chaotic mapping* to distort the payload before embedding so that it remains illegible even if the embedding method is revealed to an adversary. The payload is restorable only by supplying the correct key.

The proposed method performs better LSBR and PVD in terms of stego image *fidelity*. It alleviates the *POV* effect and withstands the χ^2 *attack* well and thus shows a better performance than the LSBR based methods. However, edge based embedding diminishes the data hiding capacity of the cover as only selective pixels are available for embedding.

Future work will focus on extending the proposed method for higher order bits in the image planes to compensate for the capacity shortfall and also on possible reduction of the technique's time complexity.

References

[1] Grant Kelly, Bruce McKenzie, "*Security, privacy and confidentiality issues on the internet*", Source:http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761937/

[2] C.V. Serdean, M. Tomlinson, J. Wade, A.M. Ambroze, "*Protecting Intellectual Rights: Digital Watermarking in the wavelet domain*", IEEE Int. Workshop Trends and Recent Achievements in IT, pp. 16-18, 2002.

[3] N.Provos, P.Honeyman, "*Hide and Seek: An Introduction to Steganography*", IEEE Security and Privacy, Vol. 1, No. 3, 2003, pp. 32–44.

[4] Ratnakirti Roy, Suvamoy Changder, Anirban Sarkar, Narayan C Debnath, "*Evaluating Image Steganography Techniques: Future Research Challenges*", International Conference on Computing, Management and Telecommunications (ComManTel 2013) [IEEE], pp. 309 – 314, January 21 - 24, 2013.

[5] D. C. Wu and W. H. Tsai, "*A steganographic method for images by pixel-value differencing*", Pattern Recognition Letters, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[6] Weiqi Luo, Fangjun Huang, Jiwu Huang, *"Edge Adaptive Image Steganography Based on LSB Matching Revisited",* IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2, June 2010, pp. 201-214.

[7] G.Karthigai Seivi, Leon Mariadhasan, K. L. Shunmuganathan, *"Steganography using Edge Adaptive Image"*, Proc. of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 1023-1027, 2012.

[8] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang , Hung-Min Sun, *"Adaptive Data Hiding in Edge Areas of Images With Spatial LSB Domain Systems",* IEEE Transactions on Information Forensics and Security, Vol. 3, No. 3, September 2008, pp.488-497.

[9] R. L. Tataru, D. Battikh, S. El Assad, H. Noura, O. Deforges, *"Enhanced Adaptive Data Hiding in Spatial LSB Domain by using Chaotic Sequences",* Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 85-88, 2012.

[10] Zhu Liehuang, Li Wenzhuo, Liao Lejian , Li Hong, *"A Novel Algorithm for Scrambling Digital Image Based on Cat Chaotic Mapping"*, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 601-605, 2006.

[11] Sahar Mazloom, Amir-Masud Eftekhari-Moghadam, *"Color Image Cryptosystem using Chaotic Maps"*, IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing, pp. 142-147, 2011.

[12] Qian-chuan Zhong, Qing-xin Zhu , Ping-Li Zhang , *"A Spatial Domain Color Watermarking Scheme based on Chaos"*, International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA), pp. 137-142, 2008.

[13] Chen Wei-bin, Zhang Xin, *"Image Encryption Algorithm based on Henon Chaotic System",* International Conference on Image Analysis and Signal Processing (IASP), pp. 94-97, 2009.

[14] John Canny,"*A computational approach to edge detection*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp.679–698, Nov. 1986.

[15] Li Bin, Mehdi Samiei Yeganeh, *"Comparison for Image Edge Detection Algorithms",* IOSR Journal of Computer Engineering, Vol. 2, Issue. 6, July-August, 2012.

[16] F. Mai, Y. Hung, H. Zhong, and W. Sze., "*A hierarchical approach for fast and robust ellipse extraction*", Pattern Recognition, Vol. 41, No. 8, pp.2512–2524, August 2008.

[17] Sergei Azernikov, "*Sweeping solids on manifolds*", Symposium on Solid and Physical Modeling,, pp.249–255, 2008.

[18] V. I. Arnold; A. Avez , "*Ergodic Problems in Classical Mechanics*", Benjamin, New York, 1968.

[19] Ron Crandall, "Some Notes on Steganography", Posted on Steganography Mailing List, 1998. Source: http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia/LINKS%20LOCALI/matrix-encoding.pdf

[20] X. Li, B. Yang, D. Cheng, and T. Zeng, "*A generalization of lsb matching"*, IEEE Signal Processing Letters, vol. 16, no. 2, pp. 69-72, 2009.

[21] Bin Li, Junhui He, Jiwu Huang, Yun Qing Shi, "*A Survey on Image Steganography and Steganalysis*", Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 2, April 2011, pp. 141-173.